

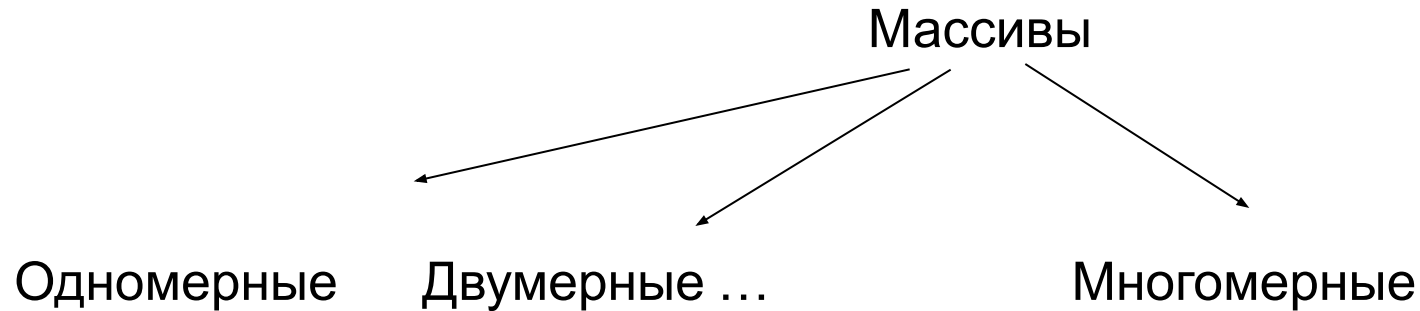
Массивы

МАССИВЫ

Массивы – это набор переменных, имеющих одно имя , один тип данных и расположенных в памяти последовательно друг за другом.

Каждая переменная массива называется **элементом массива**.

- Массивы используются для хранения ряда значений в одной переменной.
Нумерация элементов массива начинается с 0



Чаще всего используются одномерные и двумерные массивы

Массивы

Статические

Динамические

Статические массивы содержат **фиксированное количество** элементов, которое задается при описании массива

В динамических массивах **число элементов может меняться** при выполнении одной программы

Описание массивов

При описании массивов используются те же ключевые слова **Dim**, **Static**, **Private**, **Public**, как при описании обычных переменных

Dim Имя(N) As <тип данных > , где N - размерность

Dim Имя(N,M) As <тип данных > , где N - количество строк, M - количество столбцов

Нумерация элементов массивов начинается с 0

Dim A(12) As Integer	определяет элементы A(0), A(1), A(2).....A(12)
Static A(12) As Integer	
Dim B(3,3) As Integer	определяет элементы B(0,0), B(0,1), B(0,2), B(0,3),B(1,0), B(1,1) B(3,3)
Dim C(1 To 10)	определяет элементы C (1), C (2),..... ,C (10)
Dim D(1 To 3, 1 To 3)	определяет элементы D(1,1), B(1,2), B(1,3), B(2,1),B(2,2), B(3,3)

Заполнение массива

1. Оператором присваивания

```
Dim Фрукты (3) As String
```

' Здесь 3 - максимальный номер элемента

```
Фрукты(0)="Яблоки"
```

```
Фрукты(1)="Груши"
```

```
Фрукты(2)="Ананасы"
```

```
Фрукты(3)="Персики"
```

2. В цикле

```
Dim I As Integer
```

```
Dim N(10) As Integer
```

```
Randomize
```

```
For I= 0 To 10
```

```
    N(I)=Int(Rnd*10+1)
```

```
Next I
```

Пример заполнение двумерного массива

```
Private Sub CommandButton1_Click()
```

```
Dim multiarray(1, 1) As String
```

```
multiarray(0, 0) = "Здание1, Помещение1"
```

```
multiarray(0, 1) = "Здание1, Помещение2"
```

```
multiarray(1, 0) = "Здание2, Помещение1"
```

```
multiarray(1, 1) = "Здание2, Помещение2"
```

```
End Sub
```

Представление массива в памяти

Строка	Столбец 0	Столбец 1
0	Здание1, Помещение1	Здание1, Помещение2
1	Здание2, Помещение1	Здание2, Помещение2

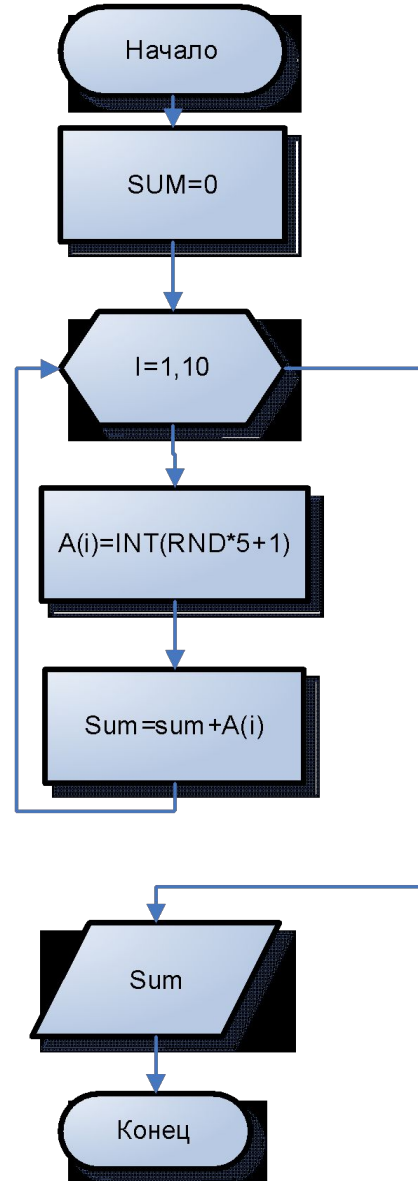
Пример заполнения и вывода массива

```
CommandButton1
Private a(10) As Integer
Private Sub CommandButton1_Click()
Dim i As Integer
For i = 0 To 10
a(i) = Int(Rnd * 20) + 1
Next i

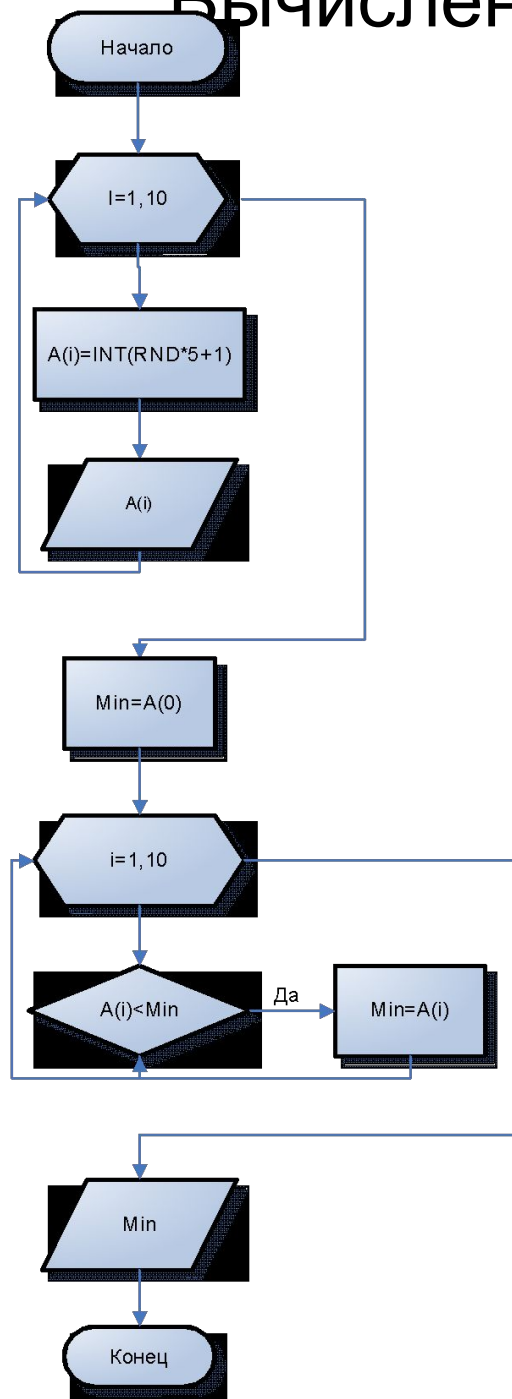
End Sub

Private Sub CommandButton2_Click()
Dim i As Integer
Dim strd As String
For i = 0 To 10
strd = strd & " " & a(i)
Next i
MsgBox strd
End Sub
```


Вычисление суммы элементов массива



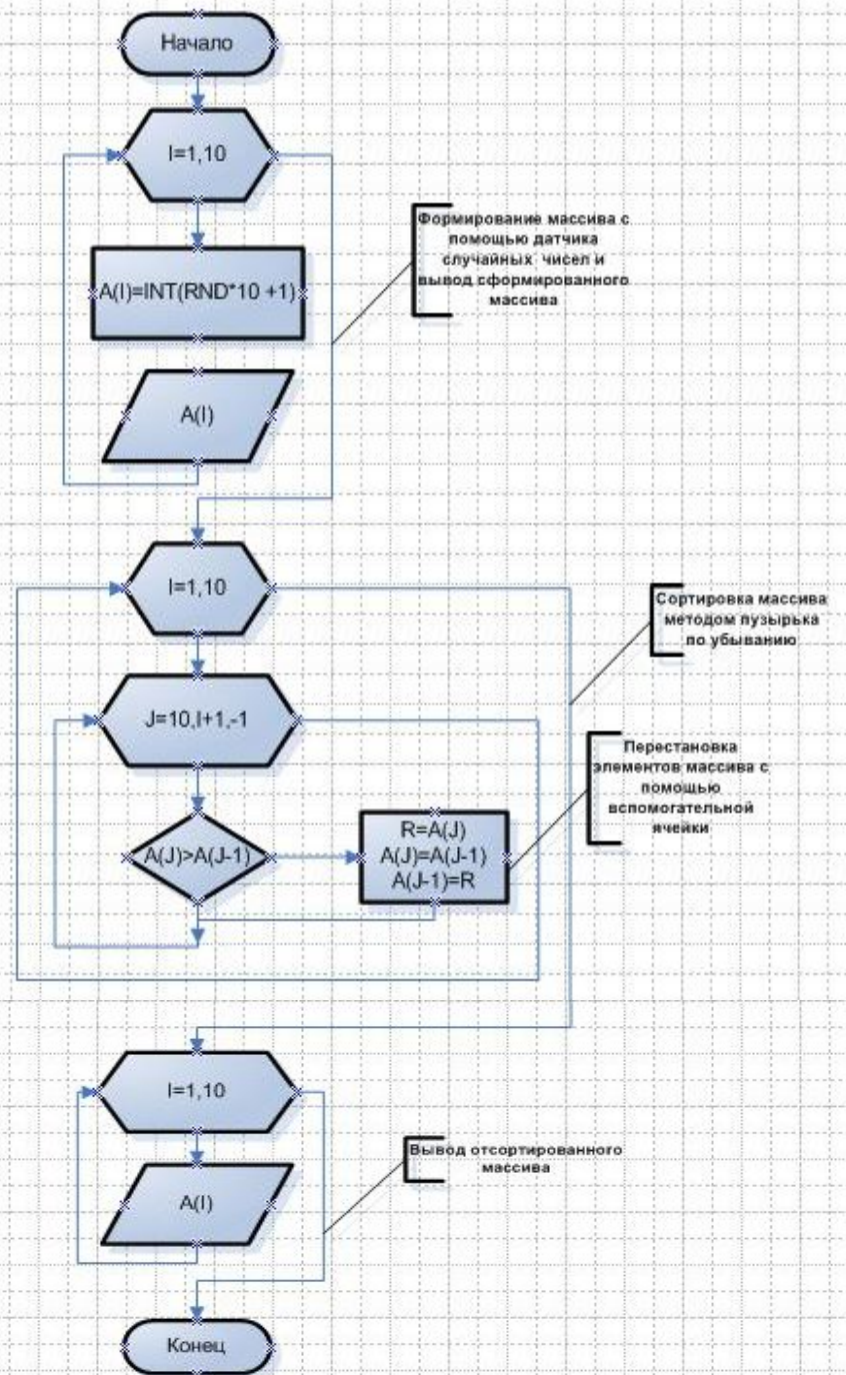
Вычисление минимального (максимального) элемента массива

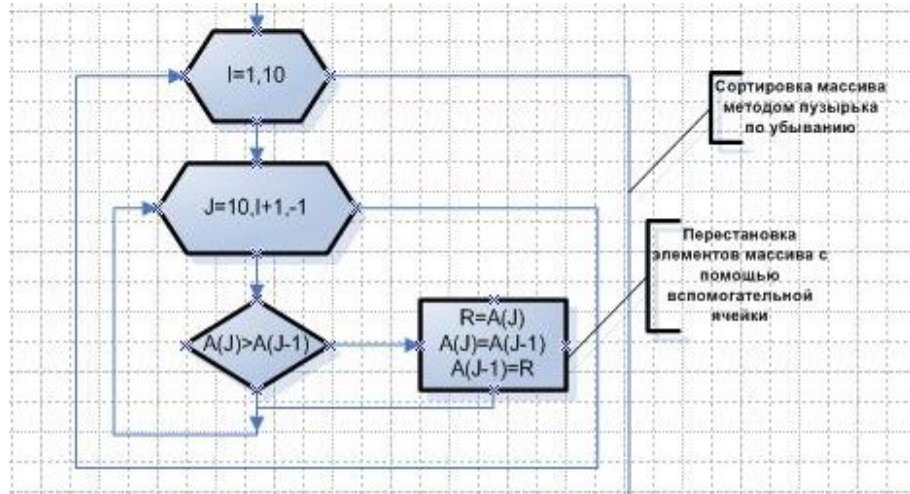


Immediate											
массив	22	17	18	9	10	24	1	23	25	22	2
min=	22										
b(1)=	17										
min=	17										
b(3)=	9										
min=	9										
b(6)=	1										
min=	1										
min=	1										
массив	13	26	24	12	29	27	2	29	11	16	24
min=	13										
b(3)=	12										
min=	12										
b(6)=	2										
min=	2										
min=	2										

Самостоятельное задание. Создать таблицу трассировки

Сортировка массива (метод пузырька)





Сортировка массива (метод пузырька)

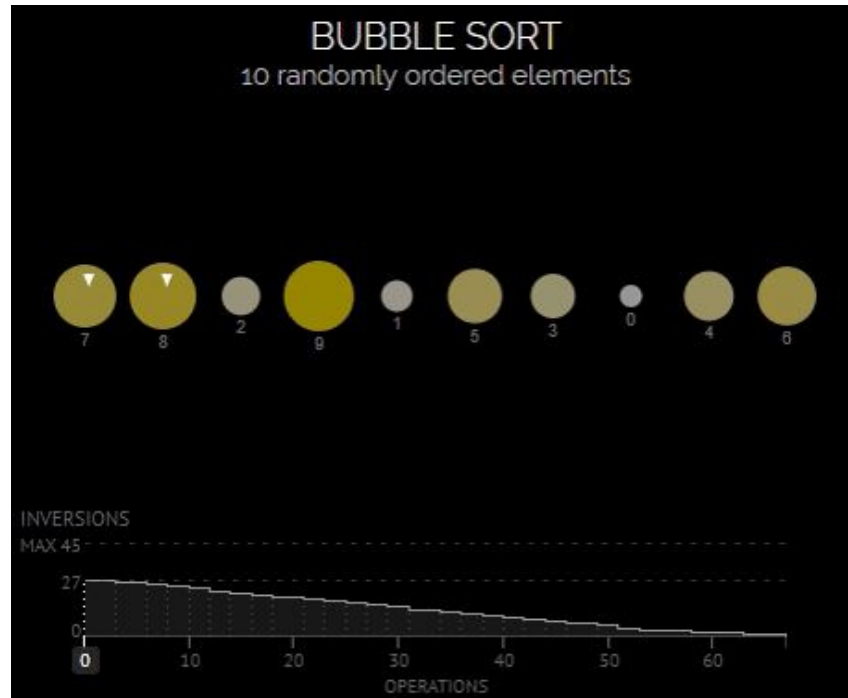
```

исходный массив  14  11  5  22  28  16
i= 0
i= 1
J= 1
b(j - 1)=14> b(j)=11      11 14
b(j - 1)=11 b(j)=14
i= 2
J= 2
b(j - 1)=14> b(j)=5      11 5 14
b(j - 1)=5 b(j)=14
J= 1
b(j - 1)=11> b(j)=5      5 11 14
b(j - 1)=5 b(j)=11
i= 3
J= 3
J= 2
J= 1
i= 4
J= 4
J= 3
J= 2
J= 1
i= 5
J= 5
b(j - 1)=28> b(j)=16
b(j - 1)=16 b(j)=28
J= 4
b(j - 1)=22> b(j)=16
b(j - 1)=16 b(j)=22
J= 3
J= 2
J= 1
новый массив  5  11  14  16  22  28
  
```

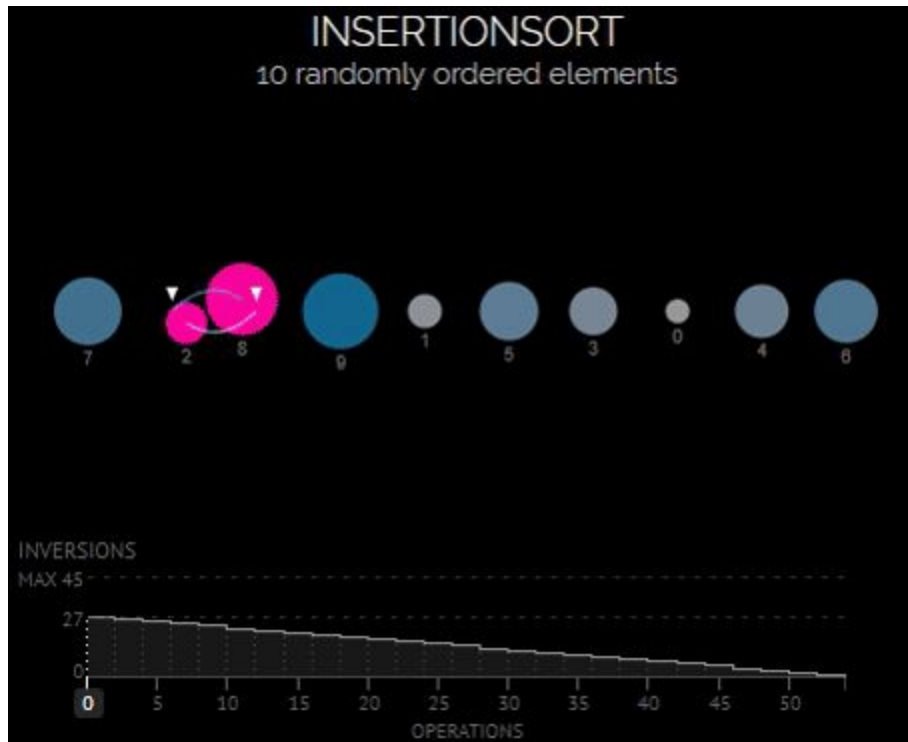
Метод пузырька(Bubble sort)

-эффективен лишь для небольших массивов

- При первом проходе по массиву элементы попарно сравниваются между собой:
 - первый со вторым,
 - второй с третьим,
 - третий с четвертым и т.д.
 - Если предшествующий элемент оказывается больше последующего, то их меняют местами.
 - самое большое число оказывается последним
- При втором проходе сравнений будет на одно меньше
- При каждом следующем проходе количество сравнений уменьшается
- Количество сравнений в каждом проходе равно $m-i$, где i – это номер прохода по массиву
- При обмене элементов массива обычно используется "буферная" (третья) переменная

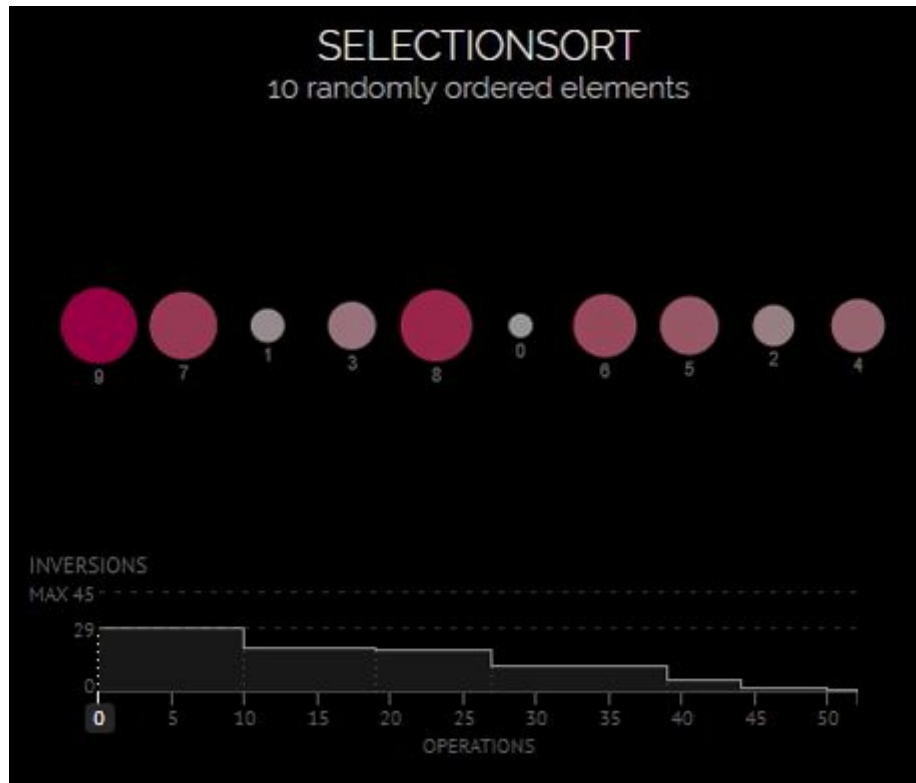


Сортировка вставками (Insertion sort)



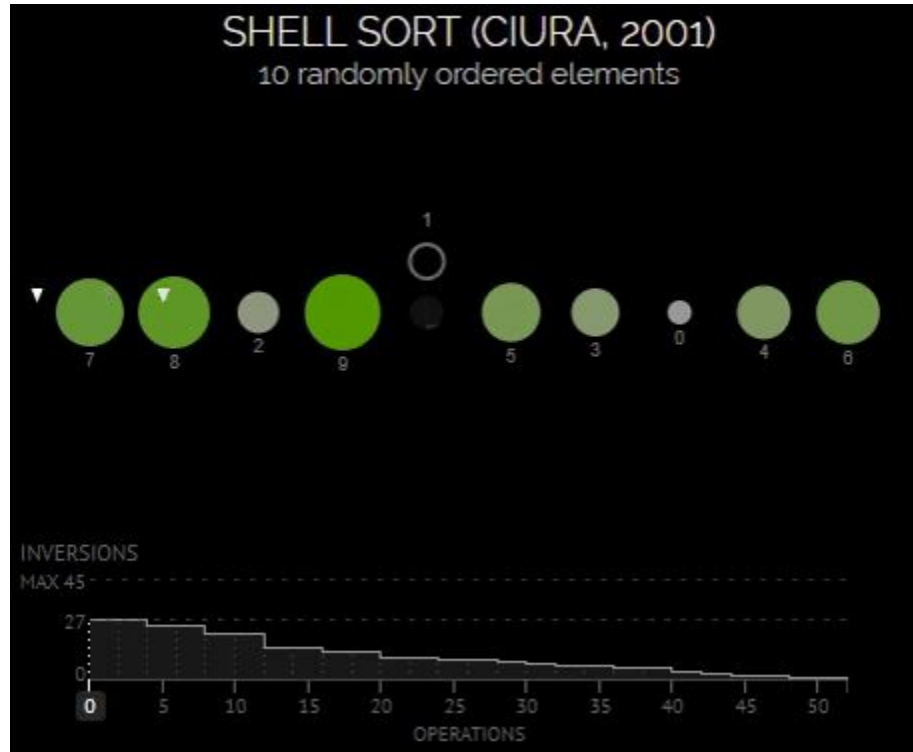
- Элементы входной последовательности просматриваются по одному
- каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов
- полезен, когда данные почти полностью отсортированы или размер данных не очень велик

Сортировка выбором (Selection sort)



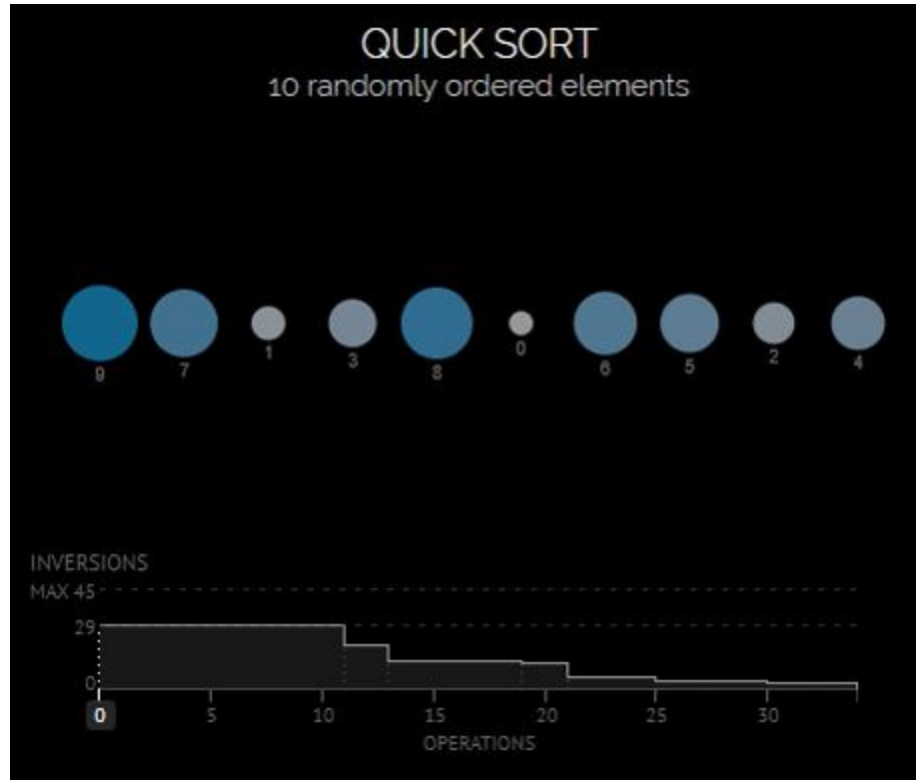
- Гибрид между пузырьковой и сортировкой вставками.
- Проходит по массиву раз за разом, перемещая одно значение на правильную позицию
- В отличие от пузырьковой сортировки, выбирает наименьшее неотсортированное значение вместо наибольшего.
- Как и при сортировке вставками, упорядоченная часть массива расположена в начале, в то время как в пузырьковой сортировке она находится в конце.

Сортировка Шелла (Shell Sort)



- Алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками.
- Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга.

Быстрая сортировка (Quick Sort)



- Выбор опорного элемента из массива.
- Перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные — после.
- Рекурсивное применение первых двух шагов к двум подмассивам слева и справа от опорного элемента. Рекурсия не применяется к массиву, в котором только один или отсутствуют элементы.

ДИНАМИЧЕСКИЕ МАССИВЫ

Бывает так, что заранее точный размер массива неизвестен, и необходимо иметь возможность изменять его размер во время исполнения программы. Чтобы изменить размер массива динамически, используется оператор **Redim**

По умолчанию, когда изменяется размер массива, все его элементы удаляются. Если необходимо **оставить текущие значения элементов массива**, то нужно использовать ключевое слово **Preserve** с оператором **Redim**, что и сделано в примере.

Для освобождения памяти, занимаемой динамическим массивом, можно использовать инструкцию **Erase**

Пример динамического массива

Не указывается количество элементов

Dim Фрукты () As String

Redim Фрукты (3)

Указание количества элементов

Фрукты(0)="Яблоки"

Фрукты(1)="Груши"

Фрукты(2)="Ананасы"

Фрукты(3)="Персики"

Переопределение количества элементов

Redim Preserve Фрукты (4)

Фрукты (4)="Сливы "

Использование функции **Array** при заполнении динамического массива

Dim Фрукты

Фрукты = Array("Яблоки", "Груши", "Сливы", "Ананасы")

Dim MyWeek

MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")

Определение размера динамического массива

Для того, чтобы определить размер массива в любой момент, используются функции

- **LBound** – нижняя граница
- **UBound** – верхняя граница

```
Private Sub CommandButton1_Click()  
Dim MyWeek, MyDay, stra  
MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")  
For i = LBound(MyWeek) To UBound(MyWeek)  
stra = stra & " " & MyWeek(i)  
Next i  
MsgBox stra  
  
MyDay = MyWeek(2)  
MsgBox "второй день недели " & MyDay  
' MyDay contains "Tue".  
MyDay = MyWeek(4) ' MyDay contains "Thu".  
MsgBox "четвертый день недели " & MyDay  
  
End Sub
```

Определение размера динамического массива

Для того, чтобы определить размер массива в любой момент, используются функции

- **LBound** – нижняя граница
- **UBound** – верхняя граница

```
Option Base 1
```

```
Private Sub CommandButton1_Click()  
Dim MyWeek, MyDay, stra  
MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")  
  
For i = LBound(MyWeek) To UBound(MyWeek)  
stra = stra & " " & MyWeek(i)  
Next i  
MsgBox stra  
  
MyDay = MyWeek(2)  
MsgBox "второй день недели " & MyDay  
MyDay = MyWeek(4)  
MsgBox "четвертый день недели " & MyDay  
  
End Sub
```

Использование цикла For ...Each

```
Private Sub CommandButton2_Click()  
Dim multiarray(1, 1) As String  
Dim vntelement As Variant  
  
multiarray(0, 0) = "Здание1, Помещение1"  
multiarray(0, 1) = "Здание1, Помещение2"  
multiarray(1, 0) = "Здание2, Помещение1"  
multiarray(1, 1) = "Здание2, Помещение2"  
For Each vntelement In multiarray  
MsgBox vntelement  
Next  
End Sub
```

Массивы и элементы управления

Список (ListBox)

Поле(TextBox)

The image shows a screenshot of a Windows application window titled "UserForm1". The window contains several controls:

- A vertical list box on the left side, which is empty. A red line points from the label "Список (ListBox)" to this control.
- Four buttons arranged in a vertical column on the right side:
 - "Заполнение Списка" (Fill List)
 - "Поиск МАХ" (Find MAX)
 - "Вычисление суммы элементов" (Calculate sum of elements)
 - "Вычисление суммы выделенных элементов" (Calculate sum of selected elements)
- Three text boxes on the right side, stacked vertically, which are currently empty. A red line points from the label "Поле(TextBox)" to the top text box.
- A "Выход" (Exit) button at the bottom right.
- A "Сортировка" (Sort) button at the bottom center, highlighted in a darker green color.

Свойства списка ComboBox и поля со списком ListBox

ListIndex	Номер текущего элемента списка
ListCount	Число элементов списка
ColumCount	Число столбцов в списке
Text	Выбранный в списке элемент
List	Элемент на пересечении строки и столбца List(1,2)
RowSource	Диапазон, содержащий элементы списка
MultiSelect	Только для ListBox
Selected	Только для ListBox
ColumnWidths	Ширина столбцов списка With ListBox1 .ColumnCount=3 .ColumnWidths="20;30;30" End With
ListRows	Устанавливает число элементов , отображаемых в раскрывающемся списке (ComboBox)

Методы

Clear	Удаляет все элементы из списка
RemoveItem	Удаляет, указанный элемент RemoveItem (5)
AddItem	Добавляет элемент